**EDA**

**Session: Data Sourcing**

EDA: It is not always possible for relevant data to be available readily in the required format for analysis and to observe patterns or insights. Also, data analysis must follow a certain step-by-step process to derive all possible insights from a data set. This step-by-step process is called as EDA.

EDA involves:

* Data Sourcing
* Data Cleaning
* Univariate Analysis
* Bivariate Analysis
* Derivation of Metrics

**Types of Data Sources**

Data can be classified into two types:

* Public data: A large amount of data collected by government or other public agencies is made public for the purpose of research. Such data sets do not require special permission for access and are, therefore, called public data.
* Private data: Private data refers to data that is sensitive to organisations and is, thus, not available in the public domain. Banking, telecom, retail, and media are some of the key private sectors that rely heavily on data to make decisions.

**Private Data**

* *Banking Data*
  + Data is sensitive.
  + Contains customer transaction details.
  + Possible decisions –
    - Giving another loan.
    - Sending collection agents.
    - Writing off the loan.
    - Promoting new products to the customer.
* *Telecom Data*
  + Optimise prepaid/post-paid plans for customers.
  + Predict and prevent customer churn.
* *Human Resources Data*
  + Predict employee attrition.
* *Retail Data*
  + Product purchasing
  + Product Stocking
  + Product Pricing
* *Media Data*
  + Advertising (TVR/TRP ratings)
* *Data Journalism*
  + Election Analysis

*The whole purpose of EDA is to analyse data without you being told what you are analysing it for.*

A person who understands the domain will produce something that is more useful than the person who understands the data. One must understand the domain to produce something useful. One must understand the data to produce something interesting.

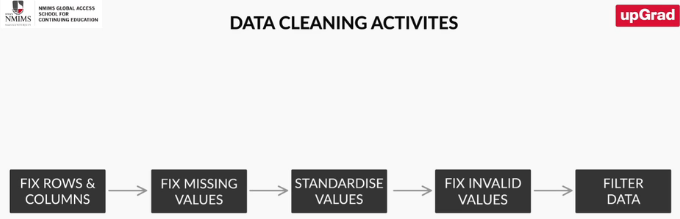
*Utility comes from a knowledge of the ability to apply.*

**Public Data**

* *Public Data Catalogues*
  + Awesome Public Datasets: <https://github.com/awesomedata/awesome-public-datasets>
  + <https://github.com/datameet>
* *Government Data Websites*
  + <https://data.gov.in>
  + <https://data.gov>
* *Census Data*
  + <https://censusindia.gov.in>

The catch with public data is to find the right data at the right time; so, as recommended by Anand, explore as many sources as you can to establish an understanding of the spectrum of data available to you.

**Session: Data Cleaning**



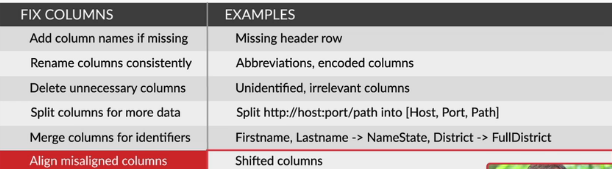
**Fix Rows and Columns**

Rows represent entity or records. Columns represent elements/ attributes within those.

Checklist for Fixing Rows

* Delete summary rows: Total, subtotal rows
* Delete incorrect rows: Header rows, footer rows
* Delete extra rows: Column number, indicators, blank rows, page number

Checklist for Fixing Columns



**Treat Missing Data**

Good method adds information, whereas a bad method exaggerates information.

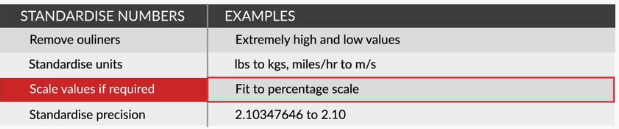
In case you can add information from reliable external sources, you should use it to replace missing values. But often it is better to let missing values be and continue with the analysis, rather than extrapolate the available information.

How to Treat Missing Values

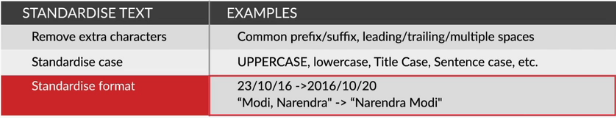
* ***Set values as missing values***: Identify values that indicate missing data and are not yet recognised by the software, e.g., treat blank strings, 'NA', 'XX', '99', etc., as missing and replace such data with a blank cell in Excel.
* ***Adding is good, exaggerating is bad***: You should try to get information from reliable external sources as much as possible; but if you cannot, then it is better to keep the missing values as such rather than exaggerating the existing rows/columns.
* ***Delete rows, columns***: Delete rows if the number of missing values is insignificant, as this would not impact the analysis. Columns could be removed if the missing values are quite significant in number.
* ***Fill partial missing value using business judgement***: These include missing time zone, century, etc., as these values are easily identifiable.

**Standardise Data**

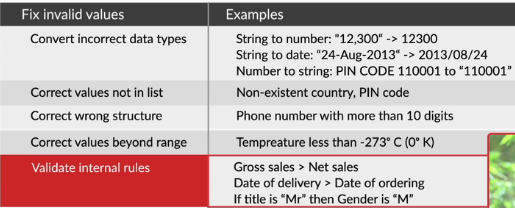
Standardise Numbers



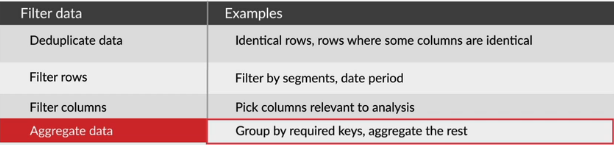
Standardise Text



**Treat Invalid Data**



**Filter Data**



**Session: Univariate Analysis**

Information about a data set can be gained simply by looking at its metadata. Metadata, in simple terms, is: 'the data that describes each variable in detail'.

Univariate analysis means that while analysing we are taking only ***one variable*** into consideration.

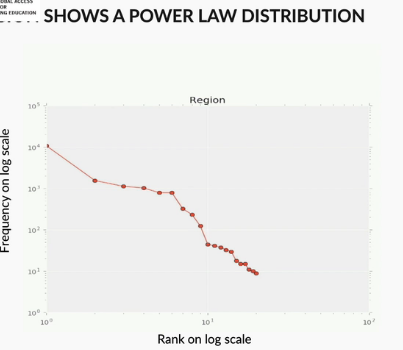
Variables can be classified as :

* **Qualitative/Categorical/Dimensional variables:**
  + **Unordered**
  + **Ordered**
* **Quantitative/ numeric variables**

**Categorical Variables – Unordered**

For a Categorical Variable column, one can do a **Rank vs Frequency Plot** – meaning which category appears how many times.

Power Law Distribution



Power Law distribution is followed when both the axis are on logarithmic scale and the graph follows a straight line decline. It is called Power Law Distribution because it follows the equation:

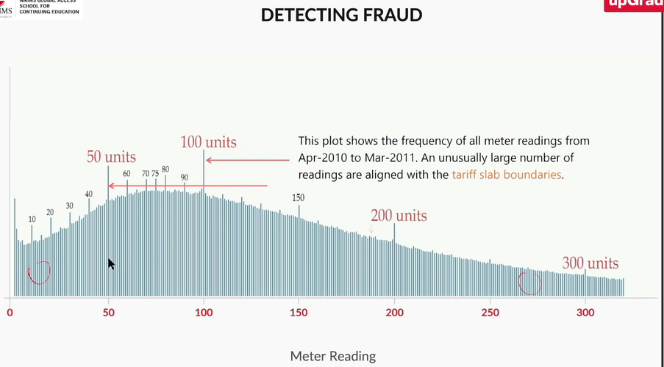
Y = aXb

It is same, as in Linear scale we have Y = aX + b.

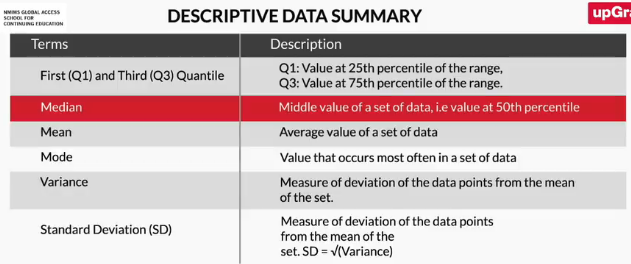
What is found is that “The ratio of the top item to the 2nd item is roughly the same as the ratio of 2nd to the 3rd; and 3rd to the 4th and so on. In other words, it forms a Geometric Progression.”

**Ordered Categorical and Continuous Variables**

Mapping a histogram for Continuous Variables can help detect aberrations or frauds by checking on the spikes.



**Quantitative Variables: Descriptive Statistics**



The mean shows the average of all the values, the median gives a typical value that could be used to represent the entire group. As a simple rule of thumb, always question someone if they use mean because the median is almost always a better measure of ‘representativeness’.

* Mode is the value that occurs most often in a set of data
* Standard deviation and interquartile difference are both used to represent the spread of the data. The interquartile difference is a much better metric than the standard deviation if there are outliers in the data; this is because the standard deviation will be influenced by outliers, whereas the interquartile difference will simply ignore them
* A box plot provides a visual representation of the spread of any data

**Segmented Univariate Analysis – I**

In segmented univariate analysis, we segment the categorical variables and then perform univariate analysis across their categories.

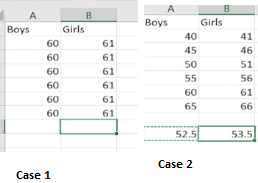
The segmentation process

1. Take raw data.
2. Group it by dimension.
3. Summarise using a relevant metric such as mean, median, etc.
4. Compare the aggregated metric across groups/categories.

**Segmented Univariate Analysis – II**

What is a T-test?

The t test tells you how significant the differences between groups are; In other words, it lets you know if those differences (measured in means) could have happened by chance.



* Both the data sets created have different distributions of the scores of boys and girls.
* In the first data set, every girl scored higher marks than every other boy. The difference in averages is still 1, but in this case, you can say that girls scored higher than boys.
* In the second data set, although the difference in averages is again 1, it is difficult to conclude that girls scored higher than boys, since the range of scores is much wider. Now, the difference is not as significant as in the previous case, since the variation in scores may cause small differences in the mean due to randomness as well.
* The above given example explains a test done to analyse whether any comparison of means done between two groups is significant or not. Such a test is known as a T-test.

**Bivariate Analysis**

Bivariate Analysis on Continuous Variables.

Bivariate analysis on Categorical Variables.

Continuous Variable Analysis:

**Correlation** is a number between -1 and 1 that quantifies the extent to which two variables ‘correlate’ with each other:

* If one variable increases with increase of the other, the correlation is positive.
* If one variable decreases with increase of the other, the correlation is negative.
* If one variable stays constant as the other varies, the correlation is zero.

Categorical Variable Analysis:

Categorical bivariate analysis is essentially an extension of a segmented univariate analysis to another categorical variable. In a segmented univariate analysis, you compare metrics such as the ‘mean of X’ across various segments of a categorical variable; e.g., mean marks are higher for students whose mothers are educated to ‘degree level and above’, the median income of educated parents is higher than that of uneducated parents, and so on.

Just as in the case of segmented univariate analysis, we need to establish whether this difference was introduced entirely by randomness. A **chi-squired** test can be used to do the same.

The usage of **chi-squared** tests to assess if the difference between any two categorical variable is due to randomness. This is done by finding the difference between the ideal case where the distribution is uniform and the current scenario:

* If the difference is small, we can attribute it to randomness
* If the difference is large, we can suggest a relation between the two variables.

**Derived Metrics**

Sometimes you would not get the most valuable insights by analysing the data available to you. Often, you need to create new variables using the existing ones to draw meaningful insights.

**Types of Derived Metrics:**

* **Steven’s typology** classifies variables into four types:
  + Nominal variables: eg Color – Red, yellow, green.
  + Ordinal variables: Things which can be ordered. Ex: Education level.
  + Interval variables: Numbers which can be subtracted. E,g, Temperature in Celsius. But these don’t represent proportionality. E.g., 20 degrees is not twice as hot as 10 degrees.
  + Ratio variables: eg., Sales,
* Numbers which wrap – longitude and latitude.
* Email address.
* URL
* Date (Season, Day of week etc) and Time (Morning/Evening/ Working hours )
* Dimensions – District/City/ TimeZone/ Ruling Party.

**Type Driven**: Just be identifying the data type of column, I will be able to identify new columns.

**Business Driven**:

* Student Pass/Fail? Based on cut off
* No. of txns in a month.
* Min avg balance maintained.

**Data Driven**:

data-driven metrics can be created on the basis of the variables present in the existing data set. For example, if you have two variables in your data set, such as 'weight' and 'height', which show a high correlation, then instead of analysing these variables separately, you can think of deriving a new metric, 'Body Mass Index (BMI)'. Once you find out the BMI, you can then categorise people easily based on their fitness level, e.g., a BMI below 18.5 should be considered in the underweight category, whereas a BMI above 30.0 would be considered obese, by standard norms. This is how data-driven metrics can help you discover hidden patterns in the data.

**PYTHON for EDA**

**SQL**

**Database**: An organised collection of a huge amount of information is called a ‘database’.

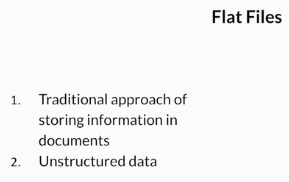
**DBMS (Database Management System)**: The term database management system comprises of database and management system. It is a software designed for managing the database. In other words, it is an application with a set of functions which help in storing, accessing, and manipulating data in an organized format.

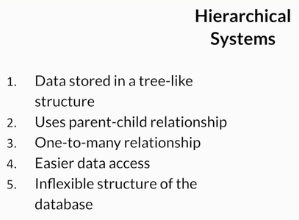
Need for DBMS

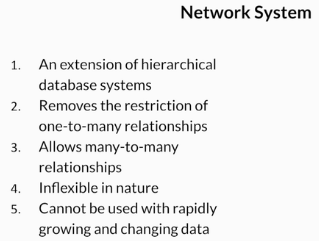
* Reduces human effort.
* Enables efficient and user-friendly data management.
* Automates and simplifies the task of maintaining data.
* Offers security and integrity of data.
* Enables data-driven decision making.

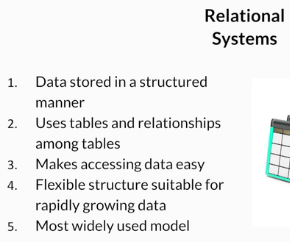
Types of DBMS

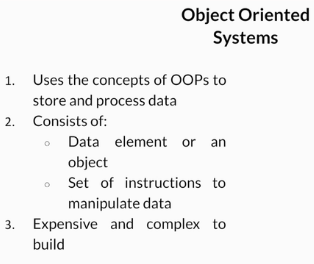
* Flat file systems
* Hierarchical database management systems
* Network database management systems
* Relational database management systems
* Object-oriented database management systems



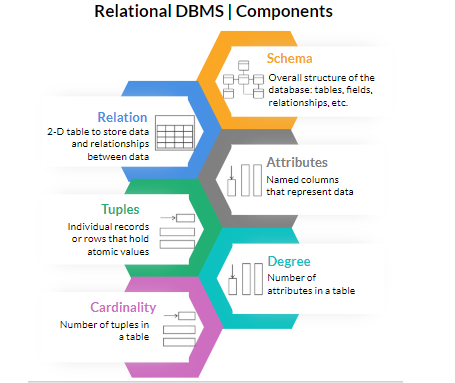








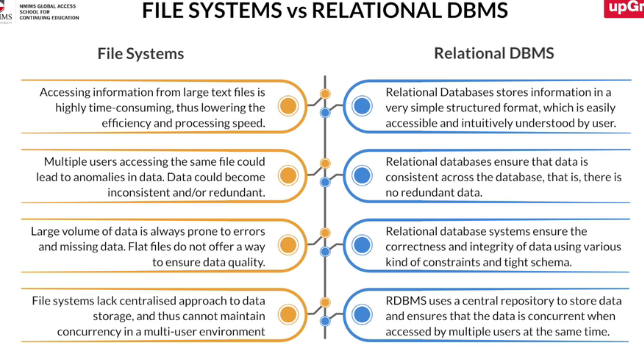
**RDBMS**



**RDBMS: Ideal Utility**

* Data growing continuously and rapidly.
* Need to keep track of historical data.
* Need for higher process speed.
* Need for securing sensitive data.
* Data and schema are subject to change with requirements.

File System vs Relational DBMS



Here are a few points of comparison:

* ***Performance***: Traditional file systems have a lower performance due to inefficient accessing techniques for large files. But an RDBMS has better efficiency due to simplified tabular structure that is easy to access.
* ***Data consistency***: All the changes made to the data are reflected consistently across tables.
* ***Data redundancy***: Data in relational databases is stored centrally and in a way that maintains consistency, thus avoiding repetition of information.
* ***Data integrity***: Data integrity is maintained by various generic checks.
* ***Data Concurrency***: A relational database allows multi-user access to data, thus maintaining concurrency across all users.

**Why SQL?**

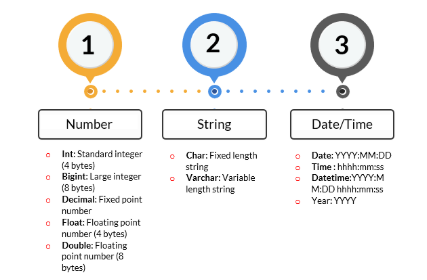
What is SQL?

Structured Query Language. It is the standard language for storing, retrieving, and manipulating data for RDBMS. It is a universal language for databases.

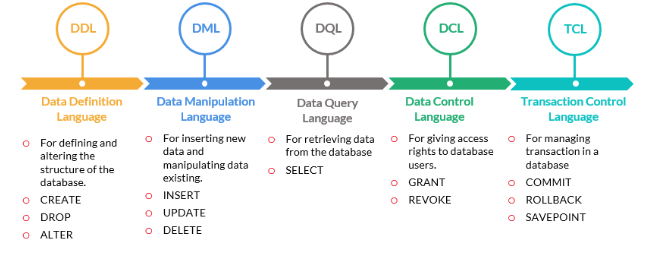
Why SQL?



**SQL DataTypes**



**SQL Commands**



**Constraints in SQL**

Constraints are rules or generic checks that can be applied to data in a table. They are used to limit the data that should be stored in a database, that is, to make sure that the data stored is valid and within acceptable ranges.

Below are the SQL Constraints:



Check example : For an Age field, check that age < 150 and >0

**Basic SQL Queries I: Extracting and Filtering**

A query in SQL is simply a call that is made in order to retrieve data from a database.

*Refer SQL Syntax.txt*

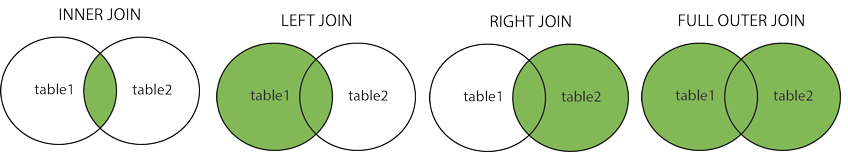
**Basic SQL Queries II: Aggregation and Sorting**

*Refer SQL Syntax.txt*

**Basic SQL Queries III: Window functions**

*Refer SQL Syntax.txt*

**Types of Joins in SQL**



**Inner Join**: Records which are present in both the tables Only are shown in the result.

**Left Join**: All Records in the left table are present and corresponding results in the right table are populated. Those values which are not present will be shown as null.

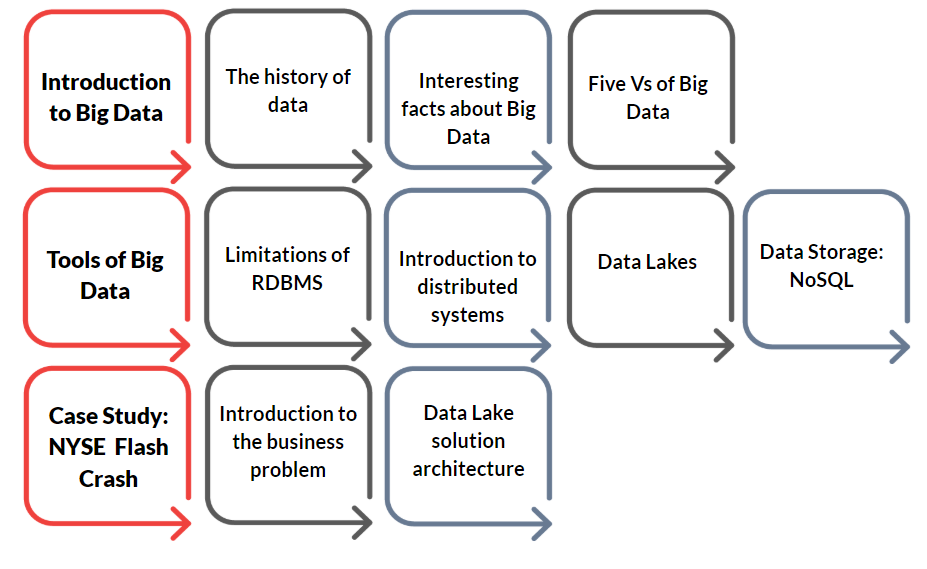
**Right Join**: All Records in the right table are present and corresponding results in the left table are populated.

**Full Join**: Result has all of the records from left table and also all of the records from right table. And if any values are not present in any of the tables, it is populated as null.

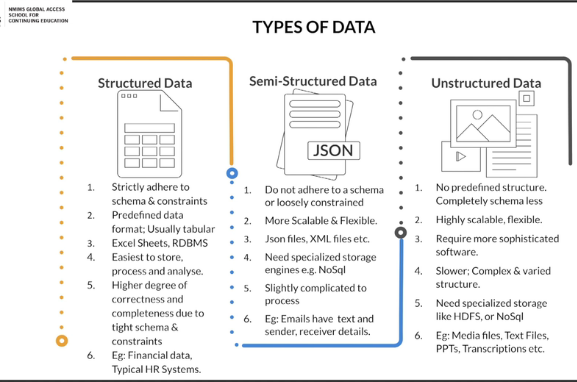
**Cross Join**:

**Big Data**

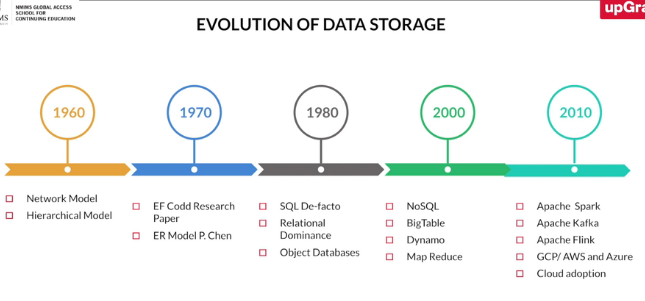
**Session: Introduction to Big Data**



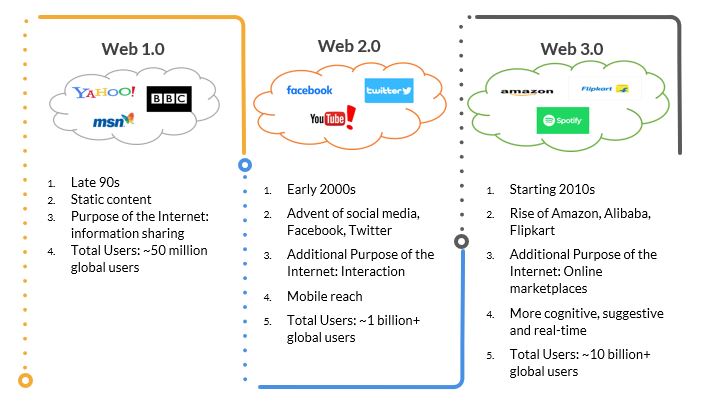
**Types of data**



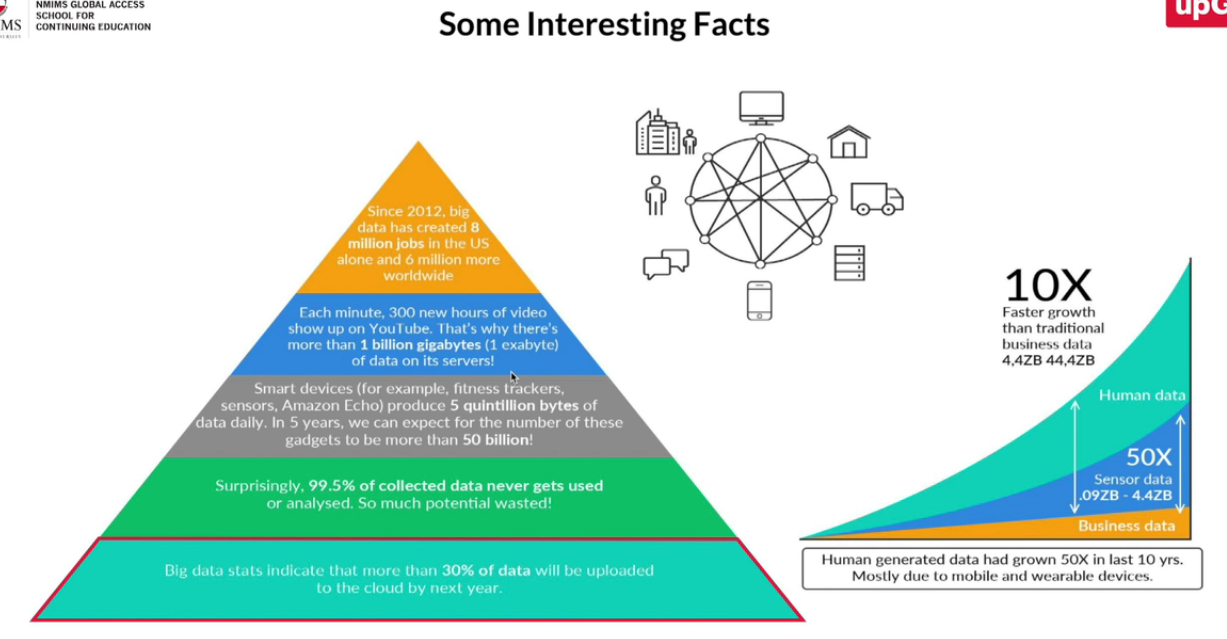
**Evolution of Data**



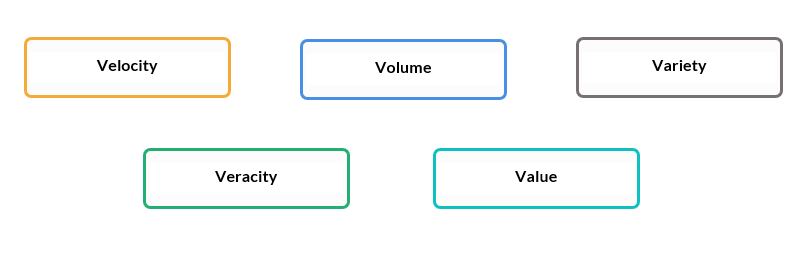
**Need for the Big Data**



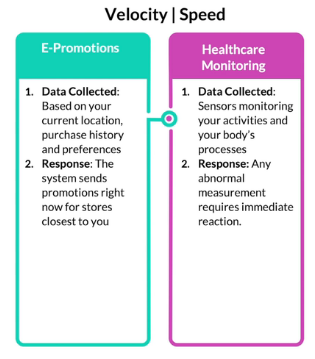
**Some Interesting Facts**

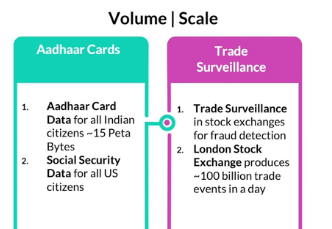
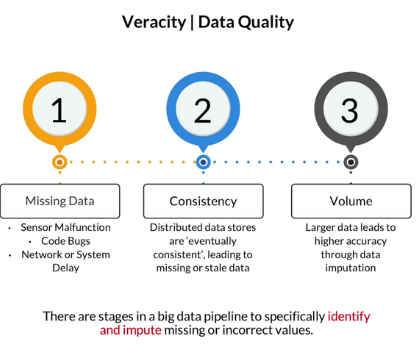
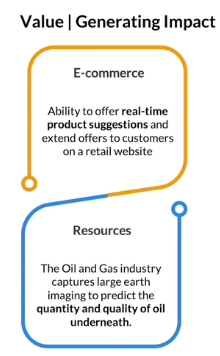


**The 5Vs of Big Data**



* **Velocity | Speed**: The speed at which data is generated and processed and is often measured in GBs/s. Data is being generated fast and needs to be processed fast**.** With such velocity of data, it is important to note that every decision that is delayed has a high opportunity cost that needs to be considered.

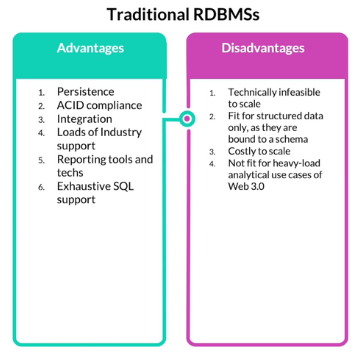


* **Volume | Scale**: The amount of data generated/processed and is measured in TBs.  
  
* **Variety | Complexity**: There are many different types of data formats, modes and structures – text, numerical, images, audio, social media data etc. Also, the data could be static or streaming.   
  
* **Veracity | Data Quality**: Veracity requires checks for Completeness, Accuracy and Timeliness of data. It also captures biases, noise and abnormality in the data.  
  
* **Value | Generating Impact**: This is the amount of tangible value that can be extracted from the data through actionable insights. A large amount of rich data makes the models more fertile for generating useful and more accurate insights.  
  

**Session: Tools for Big Data**

**Introduction to Distributed Systems**

Limitations and advantages of relational databases



ACID: Atomicity, Consistency, Integrity and Durability.

**Distributed Systems**

* A distributed system consists of at least three or more nodes connected over a network. The data is spread across these nodes in different fashion.
* It is highly scalable/ expandable as you can add nodes for increasing the net capacity.

There are 3 major Architectural patterns of Distributed Database systems:

1. **Master Slave Architecture:** 
   1. Present in RDBMS as well.
   2. One master for majority of the write operations. Parallel slaves which can perhaps do the ‘Read’ work.
   3. These have limited scalability.
   4. There may be delays between ‘Read’ and ‘Write’. That means the data that master is ‘writing’ may not be immediately available for the slaves to ‘Read’.
2. **Sharded Architecture:**
   1. Business driven design of multiple nodes.
   2. Break functional entities in different databases. E.g., Employee data in one machine and salary data in another machine.
   3. This helps to do the ‘Read’ and ‘Write’ parallelly. Hence, improves query performance drastically.
   4. However, it is very complicated to adopt. In actual life, most of the queries are made by joining different data and sharding makes it very complicated.
3. **Peer-to-peer architecture:**
   1. All nodes are equal. No master.
   2. All nodes are self-sufficient.
   3. Every node manages data that is sitting in that machine. So, even if one node goes down, other nodes continue to service.
   4. The peers may talk to each other using protocols like Gossip protocol.

In the above example, a node is defined as: 'The different systems that are connected to a network that stores/processes the data flowing through the network'.

**CAP Theorem | Use – Case – Based Choice**

* **C**onsistency
* **A**vailability
* **P**artition-tolerance

CAP Theorem is a concept that a distributed database system can only have maximum of 2 of the 3: Consistency, Availability and Partition Tolerance at any time.

**Distributed systems require partition tolerance; hence, the trade-off is between consistency and availability.**

**Consistency**: All clients have the same view of data at any given point of time. Distributed Databases are spanning across multiple nodes and it may happen that a client is attached to one node and writing some data into it and there is another client which is attached to another node of the database and reading the same data out of the database. So, there could be inconsistency if the client connects to first node versus if the client connects to another node.

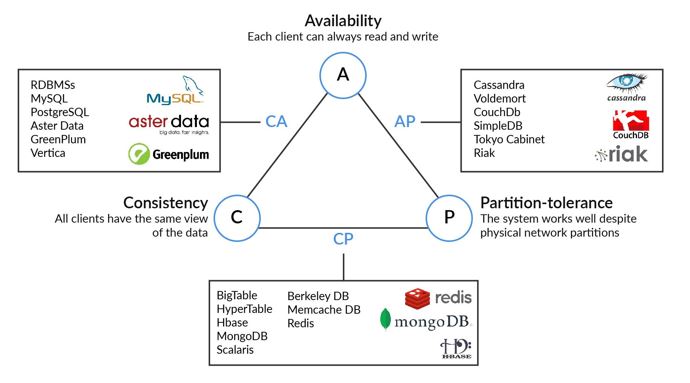
If your database is consistent, it will make sure that irrespective of the node that the client connects to, the client will receive the same data.

**Availability**: Each client can always read and write from the databases. This is typically achieved by making the system fault tolerant. E.g., Even if one node dies, the database is still up as the other nodes can service the clients.

But in case of master-slave architecture, if the master goes down, client won’t be able to read/write from the database (depending on role of master which is generally to ‘write’).

**Partition-tolerance**: This concept is unique to Distributed systems. Because Distributed systems are connected over the network, it may so happen that there is a network partition i.e., some of the nodes may get completely isolated from the rest of the nodes which leads to two different networks getting generated due to some network issue.

If your DB is partition-tolerant, that means, it can still respond to all the client queries within their own sub-networks and the client may never see that a partition has occurred and they are still able to connect and read data from the databases. Yes, they may not be able to get the right data out of it, but they are able to connect even though a set of nodes are gone into a network which cannot be reached from the other one. And the client can still read / write into the database.



**Hadoop Big Data Lake**

What is a Data Lake ?

A data lake is commonly defined as a repository that stores structured as well as unstructured data in its raw form.

Terminologies

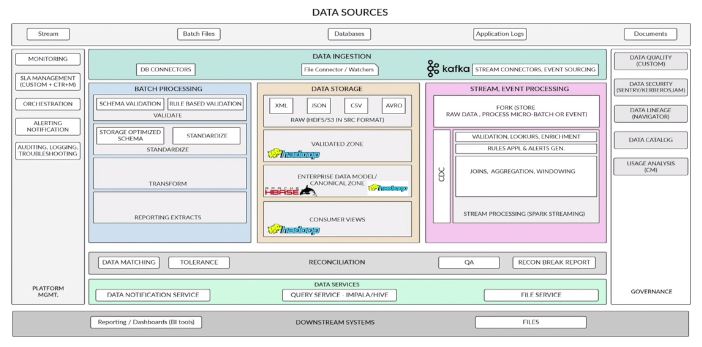
* **Database**: A database is a place meant for storing large amounts of structured data in multiple tables such that all the rows contain objects and all the columns contain attributes. The functions of a database are limited to storing/extracting data.
* **Data warehouse**: A data warehouse is built on top of multiple databases and can be used for running different analyses on any given set of data.
* **ETL**: Data is first extracted from certain sources, then transformed into an established structure and, finally, loaded., for example, data warehouses. This process of Extraction, Transformation and Loading of data is, in short, known as ETL.
* **ELT**: Data is first extracted from certain sources, then loaded and, based on requirements, transformed into the appropriate structure, for example, data lakes. This process of Extraction, Loading and Transformation of data is, in short, known as ELT.

**Layers of Data Lake:**

* Data sourcing
* Data ingestion: Process about the ways of bringing data into the data warehouse. That data could be of any type. Two types:
  + Stream ingestion.
  + Batch ingestion.
* Data storage: 4 different zones:
  + Raw: Here data is inserted as is.
  + Validated
  + Canonical mode
  + Consumer Views
* Data processing:
  + Batch Based Data processing.
  + Stream based Data processing.
* Data reconciliation
* Data services

Additional subsidiaries:

* Data governance
* Platform management



**NoSQL and NoSQL Data Stores**

NoSQL literally stands for Not only Structured Query Language, and NoSQL databases were earlier referred to as non-relational databases. These were initally designed to store unstructured data. Today, NoSQL is capable of storing all forms of data, and it is not farfetched to assume that NoSQL is the backbone of Big Data processing and analytics.

NoSQL supports unstructured data as well, unlike RDBMS which supports structured only.

Scaling is done easily – vertically + horizontally.

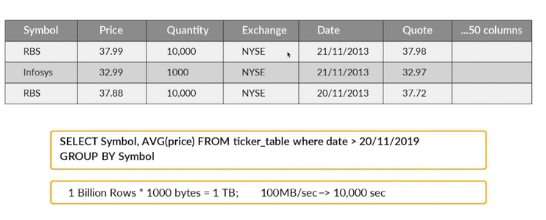
Support high variety of data.

Support high volume of data.

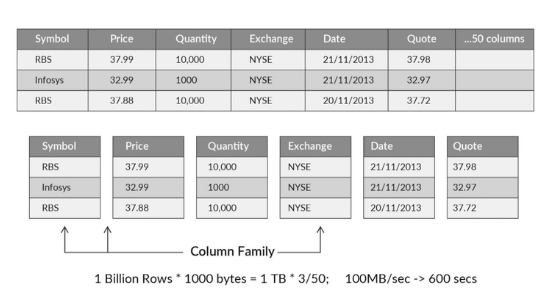
Categories of NoSQL Databases:

* Key-value database
  + Data is only stored in the form of keys and values.
* Document database
  + JSON based.
  + Schema-less.
  + Key-document.
* Graph database
  + Graph databases are used in cases where multiple relationships exist between entities. Hence, these entities are mapped with respect to one another, and each linked entity has its relationships well defined beforehand.
  + An example of graph databases would be your Netflix recommendation system, where based on different users' viewing history, relationships are created that establish if certain content liked by user A would be liked by user B.
* Columnar Database
  + Here, data is broken down into columns and a column family is made which has the columns which are often queried together.
  + This reduces access time significantly.

Row oriented data stores:



Column oriented data stores:



**One size doesn’t fit all**

Depending on the use case at hand, one may select the type of Database accordingly.